دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

پروژه پایانی درس تجهیزات IOT

گردآورندگان :

عادل کریمی ۹۹۳۶۱۳۰۵۰

دانیال توکلی ۹۹۳۶۱۳۰۱۷

امیرحسین بهرامی ۹۹۳۶۱۳۰۱۱

فهرست

سوال اول: روشن و خاموش کردن LED روی بورد وای فای را از طریق مرورگر ۳

سوال های دوم و سوم و پنجم: راه اندازی یک وب سرور روی اینترنت و روشن و خاموش کردن LED به واسطه ان ٤

سوال چهارم:خاموش و روشن کردن یک وسیله منزل از طریق اینترنت با استفاده از ماژول رله ۸

سوال ششم: دریافت فرمان از طریق تلگرام ۹

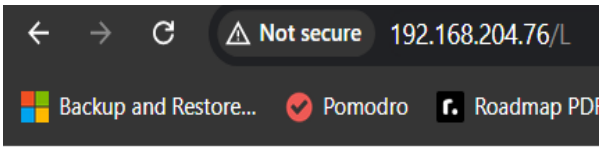
سوال اول: روشن و خاموش کردن LED روی بورد وای فای را از طریق مرورگر

تغییرات اعمال شده بر روی پروژه طبق اسلاید های درس.

```
const char *ssid = "Tartarous";
const char *password = "ajhv6004";
int led = 4;
WiFiServer server(80);
```

```
// Check to see if the client request was "GET /H" or "GET /L":
if (currentLine.endsWith("GET /H")) {
  digitalWrite(led, HIGH);  // GET /H turns the LED on
}
if (currentLine.endsWith("GET /L")) {
  digitalWrite(led, LOW);  // GET /L turns the LED off
}
```

```
Connecting to Tartarous
...
WiFi connected.
IP address:
192.168.204.76
```



Click here to turn ON the LED.
Click here to turn OFF the LED.

سوال های دوم و سوم و پنجم: راه اندازی یک وب سرور روی اینترنت و روشن و خاموش کردن LED به واسطه ان

به دلیل پیچیدگی و زمان‌بر بودن فرآیند ساخت URL، برای راه‌اندازی یک وب سرور، تصمیم گرفتیم از سرویس pythonanywhere بهره بگیریم. در این پروژه از فریمورک Django استفاده کرده‌ایم و آدرس https://danial.pythonanywhere.com/ را ایجاد کرده‌ایم. با وارد کردن این آدرس، می‌توانید به صفحه اصلی سایت دسترسی پیدا کنید و اطلاعات مورد نیاز از آنجا دریافت شود.

این وب سرور از طریقAPI های مختلف به کاربران امکان می‌دهد تا وضعیت دستگاه‌ها را بررسی و کنترل کنند و زمان‌های روشن و خاموش شدن دستگاه‌ها را برنامه‌ریزی کنند.

۱ .صفحه اصلی(Home Page)

زمانی که کاربر به URL اصلی وب سرور دسترسی پیدا می‌کند، صفحه اصلی (که به عنوان GUI.html شناخته می‌شود) نمایش داده می‌شود. این صفحه شامل یک فرم برای تنظیم زمان‌های روشن و خاموش شدن دستگاه و دکمه‌هایی برای روشن و خاموش کردن دستگاه به صورت دستی است. همچنین وضعیت کنونی دستگاه و زمان‌های تنظیم شده نمایش داده می‌شوند.

```
def firstPage(request):
    if request.method == 'GET':
        return render(request, 'GUI.html')
    else:
        return render(request, 'abc.html')
```

۲ . APIروشن کردن دستگاه(Turn On)

وقتی کاربر دکمه "Turn On" را فشار می‌دهد، یک درخواست به URL /on ارسال می‌شود. این درخواست وضعیت دستگاه را به "ON" تغییر می‌دهد و زمان‌های روشن و خاموش را به صفر برمی‌گرداند. پاسخ به کاربر شامل وضعیت جدید دستگاه یعنی("ON") و زمان‌های به‌روزرسانی شده است.

```
def get_on(request):
    global current_status, current_time_on, current_time_off
    current_status = 'ON'
    current_time_on = 0
    current_time_off = 0
    return JsonResponse({'status': current_status, 'time_on': current_time_on, 'time_off': current_time_off})
```

۳ . API خاموش کردن دستگاه(Turn Off)

با فشار دادن دکمه"Turn Off" ، یک درخواست به URL /off ارسال می‌شود. این درخواست وضعیت دستگاه را به "OFF" تغییر می‌دهد و زمان‌های روشن و خاموش را به صفر برمی‌گرداند. پاسخ شامل وضعیت جدید دستگاه یعنی("OFF") و زمان‌های به‌روزرسانی شده است.

```
def get_off(request):
    global current_status, current_time_on, current_time_off
    current_status = 'OFF'
    current_time_on = 0
    current_time_off = 0
    return JsonResponse({'status': current_status, 'time_on': current_time_on, 'time_off': current_time_off})
```

۴ . API تنظیم زمان‌بندی(Scheduling)

کاربر می‌تواند زمان‌های روشن و خاموش شدن دستگاه را از طریق فرم موجود در صفحه اصلی تنظیم کند. زمانی که فرم ارسال می‌شود، یک درخواست به URL /schedule با پارامترهای ONو OFFارسال می‌شود. این درخواست زمان‌های روشن و خاموش شدن دستگاه را تنظیم می‌کند و وضعیت دستگاه را به "Blink" تغییر می‌دهد. پاسخ شامل وضعیت جدید دستگاه یعنی("Blink") و زمان‌های تنظیم شده است.

```
def schedule(request):
    global current_time_on, current_time_off, current_status

    # Handle GET request to set time_on and time_off
    if request.method == 'GET':
        try:
            time_on_ms = int(request.GET.get('ON', '0'))
            time_off_ms = int(request.GET.get('OFF', '0'))
            current_time_on = time_on_ms
            current_time_off = time_off_ms
            current_status = 'Blink'
            response_data = {
                'status': current_status,
                'scheduling': {
                    'time_on': current_time_on,
                    'time_off': current_time_off
                }
            }
            if request.headers.get('x-requested-with') == 'XMLHttpRequest':
                return JsonResponse(response_data)
            else:
                return render(request, 'GUI.html')
        except ValueError as e:
            if request.headers.get('x-requested-with') == 'XMLHttpRequest':
                return JsonResponse({'error': 'Invalid integer values for ON and OFF parameters'}, status=400)
            else:
                return HttpResponseBadRequest('Invalid integer values for ON and OFF parameters')
    else:
        return JsonResponse({'error': 'Method not allowed'}, status=405)
```

۵ . APIبررسی وضعیت(Status Check)

برای بررسی وضعیت کنونی دستگاه و زمان‌های روشن و خاموش شدن، یک درخواست به URL /getStatus ارسال می‌شود. پاسخ شامل وضعیت کنونی دستگاه و زمان‌های تنظیم شده است. این اطلاعات به کاربر نمایش داده می‌شود تا بتواند به راحتی وضعیت دستگاه را مشاهده کند.

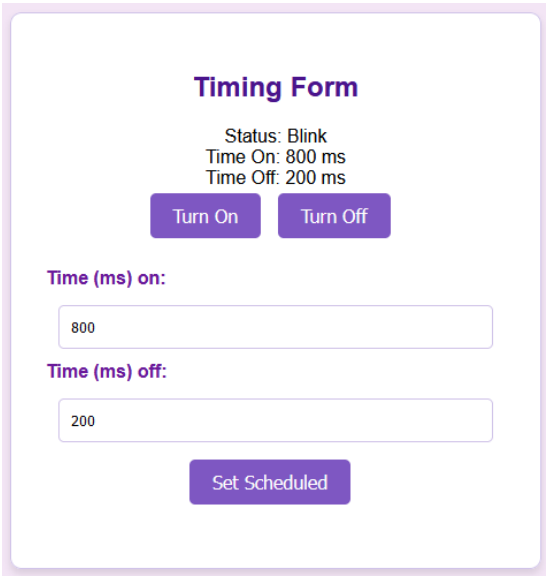
```python
def getStatus(request):
    global current_status, current_time_on, current_time_off
    if current_status == 'ON' or current_status == 'OFF':
        time_on = 0
        time_off = 0
    elif current_status == 'Blink':
        time_on = current_time_on
        time_off = current_time_off

    response_data = {
        'status': current_status,
        'scheduling': {
            'time_on': time_on,
            'time_off': time_off
        }
    }
    return JsonResponse(response_data)
```

برنامه سمت client:

با استفاده از تابع checkStatus() ، به صورت دائم به وضعیت سرور بررسی میشود و از طریق کتابخانه ی ArduinoJson، اطلاعات مربوط به وضعیت on, off, Blink و time_on, time_off دریافت میشود.

```cpp
void checkStatus() {
  client2.setInsecure();
  http.begin(client2, host);
  int httpCode = http.GET();
  if (httpCode > 0) {
    String payload = http.getString();
    DynamicJsonDocument doc(1024);
    deserializeJson(doc, payload);
    status = doc["status"].as<String>();
    time_on = doc["scheduling"]["time_on"];
    time_off = doc["scheduling"]["time_off"];
  } else {
    Serial.printf("Error on HTTP request, code: %d\n", httpCode);
  }
  http.end();
}
```

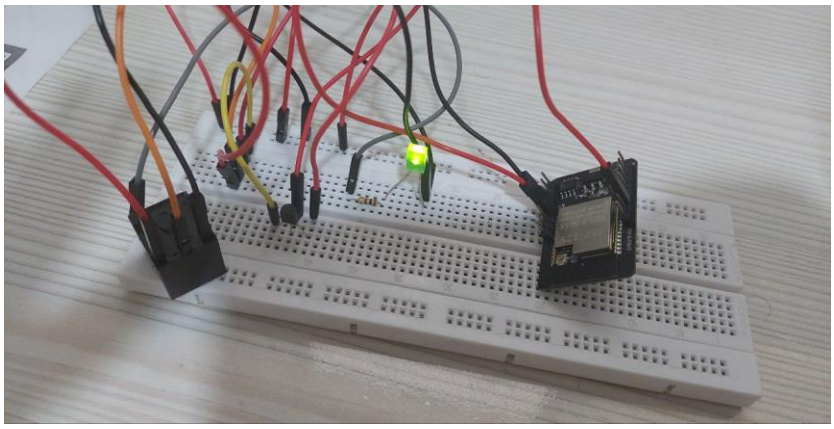
نمونه اجرا:




```
Status: Blink, time_ON: 800 ms, time_OFF: 200 ms
Status: Blink, time_ON: 800 ms, time_OFF: 200 ms
Status: Blink, time_ON: 800 ms, time_OFF: 200 ms
```

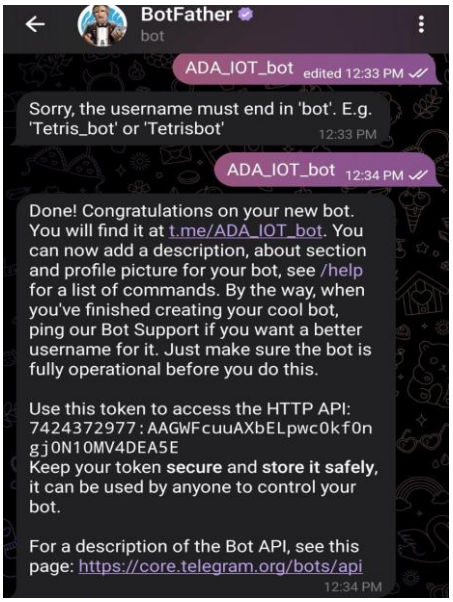
سوال چهارم:خاموش و روشن کردن یک وسیله منزل از طریق اینترنت با استفاده از ماژول رله

از برنامه ی اتصال از طریق اینترنت در قسمت های قبلی برای برنامه ریزی استفاده کردیم. بجای استفاده از ماژول رله ، از یک رله و ترانزیستور ۲N۲۲۲۲ که از قبل داشتیم ، استفاده کردیم.ابتدا تغذیه ی ماژول را با استفاده از منبع تغذیه تامین میکنیم. سپس base ترانزیستور را به پایه ESP متصل کرده و با اتصال مسیر جریان از پایه های Collector و Emitter و اتصال آن به پایه ی تغذیه ی رله (برای فعال سازی آن) ، مسیر کنترل رله را تشکیل دادیم. بعد از پیدا کردن پایه ی NO، آنرا به مقاومت و LED متصل کردیم. در ادامه همه ی GND ها را مشترک کردیم. بعد با ارسال فرمان به ESP، ترانزسیتور رله را فعال کرده و LED روشن میشود.



سوال ششم: دریافت فرمان از طریق تلگرام

از ربات BotFather@ برای ایجاد یک ربات شخصی استفاده کردیم و Token مورد نظر را داخل کد پایتون ربات و کتابخانه ی telebot استفاده کردیم.



عملکرد کلی ربات:

تعریف وظایف اصلی:

- Update Status: نمایش وضعیت کنونی دستگاه و زمان‌بندی تعیین شده.
- Turn On: روشن کردن دستگاه.
- Turn Off:خاموش کردن دستگاه.
- Set Timing:تنظیم زمان‌بندی برای روشن و خاموش شدن دستگاه.

استفاده از دکمه‌های شیشه‌ای:(Inline Keyboard)

برای سهولت کاربران، از دکمه‌های شیشه‌ای جهت انتخاب کارهای مختلف استفاده می‌شود، که این دکمه‌ها با استفاده از InlineKeyboardMarkupایجاد می‌شوند.

برقراری ارتباط با سرور از راه دور:

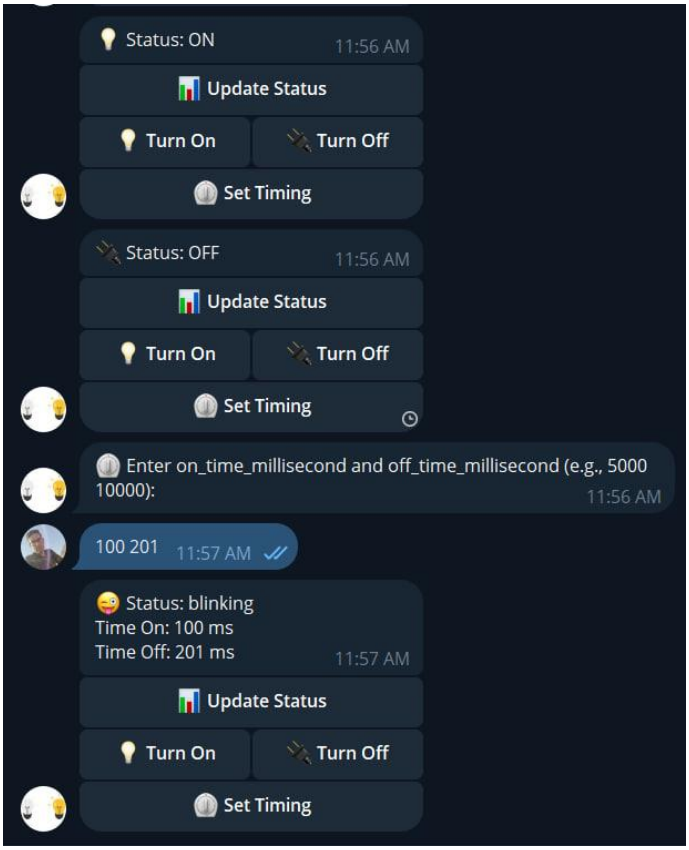
ارتباط با سرور برای ارسال درخواست‌ها بهURL های مختلف مانند روشن و خاموش کردن دستگاه و تنظیم زمان‌بندی با استفاده از requests.get() انجام می‌شود.

پردازش دستورات کاربر:

وقتی که کاربر یک دستور از طریق دکمه‌های شیشه‌ای ارسال می‌کند، این دستور توسط handle_callback پردازش می‌شود و درخواست متناظر به سرور ارسال می‌شود. برای دستورSet Timing ، کاربر باید زمان‌های روشن و خاموش کردن دستگاه را به صورت میلی‌ثانیه وارد کند، که این داده‌ها نیز به سرور ارسال می‌شود.

پیام های خطا:

در صورتی که هنگام ارسال یا دریافت اطلاعات مشکلی به وجود بیاید، پیام‌های خطا به کاربر نمایش داده می‌شود تا اوضاع بهتر مشخص شود.



ساختار کلی کد

کد ربات شامل بخش‌های زیر است:

وارد کردن کتابخانه‌ها:

- telebot برای ساخت ربات تلگرام.
- Requests برای ارسال درخواست‌های HTTP.
- telebot.types برای استفاده از دکمه‌های شیشه‌ای (Inline Keyboard).

تعریف متغیرهای اصلی:

- توکن ربات تلگرام.
- آدرس‌های URL برای ارسال درخواست‌های مربوط به روشن و خاموش کردن دستگاه، تنظیم زمان‌بندی و به‌روزرسانی وضعیت.

تعریف ربات:

ایجاد یک نمونه از کلاس TeleBot با استفاده از توکن.

ایجاد دکمه‌های شیشه‌ای:

تابع create_inline_keyboard برای ایجاد و بازگرداندن دکمه‌های شیشه‌ای.

مدیریت درخواست‌های کاربر:

تابع handle_callback برای پردازش درخواست‌های کاربر بر اساس داده‌های برگشتی از دکمه‌های شیشه‌ای.

پردازش زمان‌بندی:

تابع process_timing_step برای دریافت و پردازش زمان‌های وارد شده توسط کاربر جهت تنظیم زمان‌بندی روشن و خاموش شدن دستگاه.

مدیریت دستورات start وhelp :

تابع handle_start_help برای نمایش پیام خوشامدگویی و دکمه‌های شیشه‌ای به کاربر.

شروع به کار ربات:

فراخوانی bot.polling() برای شروع دریافت و پردازش پیام‌ها.

توضیح توابع و فرآیندها

تابعcreate_inline_keyboard:

این تابع یک InlineKeyboardMarkup ایجاد می‌کند و دکمه‌های مختلفی را به آن اضافه می‌کند:

- دکمه "Update Status" برای به‌روزرسانی وضعیت دستگاه.
- دکمه "Turn On" برای روشن کردن دستگاه.
- دکمه "Turn Off" برای خاموش کردن دستگاه.
- دکمه "Set Timing" برای تنظیم زمان‌بندی.

تابعhandle_callback:

این تابع برای پردازش درخواست‌های کاربر زمانی که یکی از دکمه‌های شیشه‌ای فشرده می‌شود استفاده می‌شود. برای هر درخواست، یک درخواست HTTP به URL متناظر ارسال می‌شود و پاسخ دریافت شده به کاربر نمایش داده می‌شود.

```
@bot.callback_query_handler(func=lambda call: True)
def handle_callback(call):
    if call.data == 'update_status':
        response = requests.get(API_URL_UPDATE_STATUS)
        data = response.json()
        bot.send_message(call.message.chat.id, f"📊 Status: {data['status']}\n"
                                               f"Time On: {data['scheduling']['time_on']} ms\n"
                                               f"Time Off: {data['scheduling']['time_off']} ms",
                         reply_markup=create_inline_keyboard())
    elif call.data == 'turn_on':
        response = requests.get(API_URL_TURN_ON)
        data = response.json()
        bot.send_message(call.message.chat.id, f"💡 Status: {data['status']}\n", reply_markup=create_inline_keyboard())
    elif call.data == 'turn_off':
        response = requests.get(API_URL_TURN_OFF)
        data = response.json()
        bot.send_message(call.message.chat.id, f"🕯️ Status: {data['status']}\n", reply_markup=create_inline_keyboard())
    elif call.data == 'set_timing':
        msg = bot.send_message(call.message.chat.id,
                               "⏲️ Enter on_time_millisecond and off_time_millisecond (e.g., 5000 10000):")
        bot.register_next_step_handler(msg, process_timing_step)
```

تابعprocess_timing_step :

این تابع زمان‌های وارد شده توسط کاربر را پردازش می‌کند و آن‌ها را به سرور ارسال می‌کند تا زمان‌بندی روشن و خاموش شدن دستگاه تنظیم شود. در صورت موفقیت‌آمیز بودن درخواست، وضعیت جدید به کاربر نمایش داده می‌شود.

تابعhandle_start_help :

این تابع پیام خوشامدگویی و دکمه‌های شیشه‌ای را به کاربر نمایش می‌دهد.

```
@bot.message_handler(commands=['start', 'help'])
def handle_start_help(message):
    """Sends a message with the inline keyboard to the user."""
    bot.send_message(
        message.chat.id,
        "Welcome! Here are the options:",
        reply_markup=create_inline_keyboard()
    )
```

شروع به کار ربات:

با فراخوانی bot.polling()، ربات شروع به گوش دادن به پیام‌های دریافتی و پردازش آن‌ها می‌کند.

لینک گیتهاب پروژه:

https://github.com/adel٤٧٨٠/IOT-Final

لینک وب سرور:

https://danial.pythonanywhere.com/

لینک بات: ابتدا فایل برنامه + فیلترشکن اجرا شود

http://t.me/ADA_IOT_bot